CLAIMS

Having thus described our invention, what we claim as new and desire to secure by Letters Patent is as follows:

1. A method for storing information in a recoverable manner on an untrusted system, comprising:

sending, by a client, a request to a recovery server for recovery of a failed database;

determining whether said request is legitimate;

based on said determining, sending an old local key to the client;

decrypting by said client the failed database with the old local key, to recover the failed database; and

re-encrypting the recovered database with a new local key.

2. The method of claim 1, further comprising:

verifying whether a key database identification has been tampered with.

3. The method of claim 1, wherein said database is associated with content which is purchased from a content owner and stored, along with a keyword or codeword, on the database of the client.

11. The method of claim 9, further comprising incrementing a counter periodically and stored to the nonvolatile location, such that a restored value will be saved with a wrong key.

12. The method of claim 1, wherein a unique key is produced by using a combination of a local storage and a nonvolatile location of a computer system of said client.

13. The method of claim 1, wherein the database is encrypted with a local key that is used with the database, and the local key is encrypted such that it is decryptable only by the recovery server.

14. The method of claim 13, wherein the recovery server uses public key cryptography to decrypt the local key.

15. The method of claim 1, wherein said recovery server automatically provides said key at a first request thereof.

16. The method of claim 15, wherein if the new local key is not correct, the client extracts the encrypted old local key, and sends it to the recovery server and the recovery server judges whether to allow the recovery.

17. The method of claim 16, wherein if the recovery should be allowed, said method further comprising:

decrypting the key and sending it back to the client for decrypting the information and re-encrypting the decrypted information with a new valid local key.

18. The method of claim 1, wherein the data in a non-volatile area of a machine of said client is changed every time a count changes, such that said local key also changes.

19. The method of claim 1, wherein random keys are used to encrypt the data, and the local key is used to encrypt the random keys.

20. The method of claim 1, wherein counters are kept in records of the database, and the local key is used to encrypt the counters.

21. The method of claim 1, wherein the request includes a header and a body, all but a first portion of the header being encrypted with a local encryption key.

22. The method of claim 21, wherein a cleartext portion of the header contains a database ID which is unique among all the users, said ID serving as an identification during recovery.

23. The method of claim 22, wherein a second portion of the header is a combined item which contains the local encryption key and the database ID, encrypted with the recovery center's public key, such that this item is only in the clear regarding the local key database key.

24. The method of claim 23, wherein a remaining portion of the key database includes fields which are encrypted with the local key, including the rest of the header, which includes the key database ID and a codeword, said fields serving to check whether the key is calculated correctly or the system has been modified or tampered with.

25. The method of claim 24, wherein an entirety of said body is encrypted with the local encryption key.

26. The method of claim 25, wherein a decryption key for the database cannot be reconstructed locally and must explicitly be recovered, such that a client application program extracts the second field from a cleartext key database header,

said second field containing a lost key and is encrypted with the recovery center's public key, so that only the recovery server can decrypt it using its private key, said client sending the second field to the recovery server, for checking the legality of this action, decrypting the key and returning the key so that a client application decrypts the old key database, generates a new key and sets the system parameters and encrypts the key database.

27. The method of claim 1, wherein the recovery server determines whether or not to automatically grant the recover operation based on any of whether a normal user upgrade is due, and a predetermined time period has elapsed between user recovery of a failing machine.

28. The method of claim 1, wherein operator-assisted recovery is performed by resetting certain parameters in the decision logic and making the client re-request recovery.

29. A method of allowing recovery of a proprietary database, comprising:

detecting, by a recovery server, a request to restore a database;

determining, by the recovery server, whether the request is legitimate by verifying an identification (ID) of a key database identification included in the request of the user;

based on the ID matching a predetermined ID, then applying a recovery decision logic, and granting the restore by the recovery server;

forwarding a local key that the database was incorporated with to a user;

using the local key, calculating a new local key by decrypting the database with the local key, such that the database is re-encrypted with the new local key.

30. The method of claim 29, wherein operator-assisted recovery is performed by resetting certain parameters in the decision logic and making the client re-request recovery.

31. A system for storing information in a recoverable manner on an untrusted system, comprising:

means for sending, by a client, a request to a recovery server for recovery of a failed database;

means for determining whether said request is legitimate;

based on an output from said means for determining, means for sending an old local key to the client;

means for decrypting by said client the failed database with the old local key, to recover the failed database; and

means for re-encrypting the recovered database with a new local key.

32. The system of claim 31, wherein operator-assisted recovery is performed by resetting certain parameters in the decision logic and making the client re-request recovery.

33. A system of allowing recovery of a proprietary database, comprising:

means for detecting, by a recovery server, a request to restore a database;

means for determining, by the recovery server, whether the request is legitimate by verifying an identification (ID) of a key database identification included in the request of the user;

means for applying a recovery decision logic based on the ID matching a predetermined ID, and for granting the restore by the recovery server;

means for forwarding an old local key that the database was incorporated with to a user;

means, using the old local key, for calculating a new local key by decrypting the database with the old local key, such that the database is re-encrypted with the new local key.

34. The system of claim 33, wherein operator-assisted recovery is performed by resetting certain parameters in the decision logic and making the user re-request recovery.

35. A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of storing information in a recoverable manner on an untrusted system, comprising:

sending, by a client, a request to a recovery server for recovery of a failed database;

determining whether said request is legitimate;

based on said determining, sending a local key to the client;

decrypting by said client the failed database with the local key, to recover the failed database; and

re-encrypting the recovered database with a new key.

36. A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of allowing recovery of a proprietary database, comprising:

detecting, by a recovery server, a request to restore a database;

determining, by the recovery server, whether the request is legitimate by verifying an identification (ID) of a key database identification included in the request of the user;

based on the ID matching a predetermined ID, then applying a recovery decision logic, and granting the restore by the recovery server;

forwarding a local key that the database was incorporated with to a user;

using the local key, calculating a new local key by decrypting the database with the local key, such that the database is re-encrypted with the new local key.

4. The method of claim 3, wherein the client can access the recovery server with the keyword to restore the database.

5. The method of claim 1, wherein a unique key is provided for each piece of content in said database and an overall key is provided for the entire database.

6. The method of claim 1, wherein the keys protect the content of the database, the keys being dependent upon unique characteristics of a hardware component associated with said database.

7. The method of claim 1, wherein the keys are based on at least one of a processor identification, a particular sector of a system file and random data stored in a non-volatile area of a computer system of said client.

8. The method of claim 7, wherein said random data includes values placed in a secret location in the system, said secret location including any of a system's basic input/output system (BIOS), a nonvolatile RAM (NVRAM), and a hard disk.

9. The method of claim 1, wherein the local keys are dependent on a value in at least one secret location which changes every time a predetermined action occurs.

10. The method of claim 9, further comprising storing a counter in the secret, nonvolatile location and incremented.